Smart-Net Gateway

Installation and User's Manual

Table of Contents

Overview 5
Features 5
Packing Materials 5
Contact Information 5
Installation 6
Connecting the Smart-Net Gateway 6
Connecting Smart-Net Devices 6
Configuring an IP Address 7
Using the IPGatewaySetup Java Utility 7
Using a Web Browser 8
Web Interface 11
Logging in 11
Changing Network Settings 12
Adjusting the Date and Time 13
Setting the Time Zone 14
Manually Setting the Date and Time 14
Automatically Setting the Date and Time 15
Rebooting the Smart-Net Gateway 15
Changing the System Password 15
Viewing the Connected Devices History 16
Changing FTP Logging Settings 17
Changing Dial Up Networking Settings 17
XML-RPC 18
Overview 18
Accessing XML-RPC Methods 18
Common Field Descriptions 19
Timestamp 19
Status 19
ROMID 20
ParentID 20
UserID 20

Smart-Net Gateway XML-RPC Methods 20
SmartNet.Status 20
SmartNet.Time 21
SmartNet.Find 21
SmartSenseTH.Read 22
SmartSenseTH.Write 23
SmartWatt.Read 24
SmartWatt.Write 27
SmartPDU.Read 28
SmartPDU.Map 29
SmartPDU.Write 29
Additional Resources 30
FTP Logging 31
Overview 31
Configuring and Testing 31
Enabling/Disabling 32
Server Host 33
Server Username/Password 33
Remote Log Directory 33
Reading and Upload Intervals 33
Alerting Options 33
Enabling/Disabling 33
From Address 34
To Address 34
Message Type 34
Log File Formats 34
Common Fields 35
SmartSenseTH 36
SmartWatt 37
SmartPDU 38
Smart-Net 39
Exception 39
Test 40

Intervals and Memory Usage 40
Attaching a Modem/PPP 40

Overview

Features

The Smart Works Smart-Net Gateway provides connectivity to your Smart-Net network from TCP/IP networks via an Ethernet connection and a static IP address. This enables you to manage and communicate with an entire Smart-Net network from a single static IP address.

The Smart-Net Gateway currently supports the following features:

- Plug and Play self discovery of new devices
- Flexible wiring topologies up to 1000' with Category 5 cable
- Intuitive web interface for configuration
- XML based procedure calls (XML-RPC)
- Automatic device monitoring via FTP
- (Optional) Dial Up Networking Support via an external modem

Packing Materials

The Smart-Net Gateway is shipped with the following materials:

- Smart-Net Gateway
- Power supply
- CD-ROM (which includes this users manual and additional resources)

Contact Information

If you wish to contact Smart Works, Inc., you may do so using one of the following methods:

- **E-Mail** – info@smart-works.com
- **Telephone** – (253) 735-0552
- **Fax** – (253) 735-0562
- **Web** – http://www.smart-works.com

Installation

Connecting the Smart-Net Gateway

The Smart-Net Gateway features the following connections (Figure 1):

- One Console Serial Port
- One External Serial Port
- One 5V DC, 300mA Power Input Jack
- One RJ-12 Smart-Net Device Port
- One RJ-45 Ethernet Port

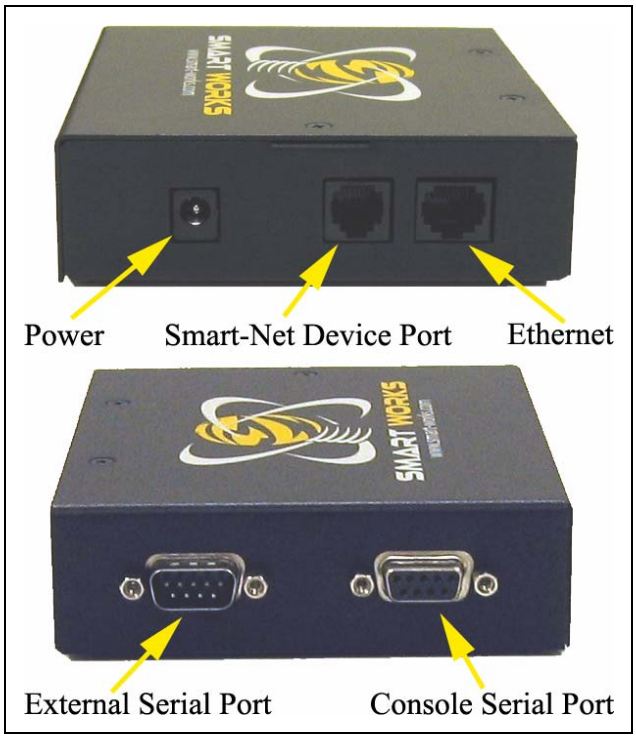

Figure 1. Smart-Net Gateway Connections

To connect the Smart-Net Gateway, perform the following steps:

1. Place the Smart-Net Gateway at the desired location
2. Use Category-5e Cable to attach the Smart-Net Gateway to the network via the Ethernet port.
3. Connect the power cord and plug the transformer into a wall-socket.

Connecting Smart-Net Devices

The Smart-Net Gateway has a built-in RJ-11 connector that controls a Smart-Net Device Network. Every Smart-Net device has two RJ-11 connectors to allow for daisy chaining of devices. Simply connect one of the devices on the Smart-Net Network to the RJ-11

port on the Smart-Net Gateway. The devices will automatically be detected and no additional configuration is necessary.

There are a few points to keep in mind when installing a Smart-Net Network:

- **Polarity** – The Smart-Net network uses standard RJ-11 connectors to make it easy and convenient to wire and extend. If standard telephone cable is used, it is important to ensure that it is not of a "flipped" type. The cable must be "straight-through", that is pin 1 on one end of the cable must conduct directly to pin 1 on the other end of the cable. The same goes for all the pins on the cable.
- **Topology** – The recommended topology of wiring devices in the Smart-Net Network is in a "daisy chain" fashion. That is, each additional device should extend from the last device on the network. Wiring the network this way will ensure reliable communication and the longest distances. Branching off to other groups of Smart-Net devices from one point will cause signal errors and greatly deteriorate the reliability and maximum distance of the network.
- **Cable Grade** – Using Category 5 or Category 6 wire will support the longest distances with minimal communication errors. However, if the network is very short, lower grade cable may be used.

Configuring an IP Address

When the Smart-Net Gateway has been powered up for the first time, it will attempt to retrieve an IP Address with DHCP. If a DHCP server is unreachable, it will set a default IP Address of 169.254.x.x (where x.x is the last two bytes of the Smart-Net Gateway's MAC Address) with subnet mask 255.255.0.0. The device will also listen to a Multicast group at 224.0.10.10 on port 6082.

Using the IPGatewaySetup Java Utility

The supplied Java program, "IPGatewaySetup", will configure Smart-Net Gateways via the Multicast group described above. When changes are made to a Smart-Net Gateway with this utility, DHCP will be automatically disabled to ensure your configuration changes take effect. To configure the IP Address with the IPGatewaySetup Java Utility, perform the following steps:

1. Connect the Smart-Net Gateway to the same networking hardware as a PC with the Java Virtual Machine runtime environment installed.
2. Power on the Smart-Net Gateway and wait about one minute for it to initialize.
3. Run the program "IPGatewaySetup" on the PC.
    - For example, on Windows, open a command prompt, change to the directory where "IPGatewaySetup" is installed, and type:

      ```
      java –cp . IPGatewaySetup
      ```

4. The program will connect to the Multicast group, search for all Smart-Net Gateways, and display them to the screen with the device's MAC address, IP Address, and Subnet Mask (Figure 2).

C:\WINDOWS\system32\cmd.exe - java -cp . IPGatewaySetup
C:\SmartNet>java -cp . IPGatewaySetup
Searching for Smart Net IP Gateways...
1. Mac: 00:60:35:01:a0:2f, IP:192.168.0.200, Subnet Mask: 255.255.255.0
Which device to configure [1] (q to quit, r to refresh)?

Figure 2. IPGatewaySetup Java Program

5. Select the device you are configuring and press enter.
6. Type in the new IP Address for the device and press enter.
7. Type in the new Subnet Mask for the device and press enter.
8. Type in the new Default Gateway for the device and press enter. If there is no gateway address, leave the field blank and press enter.
9. The program will send the request and refresh the list of Smart-Net Gateways found.
10. Proceed with the configuration from the "Web Interface" section of this manual.

Using a Web Browser

To configure the IP Address with a Web Browser by changing the PC's network settings:

1. If you have a DHCP server on your network, check the server logs or use the Java utility (described above) to determine the Smart-Net Gateway's IP Address. If you do not have a DHCP server, determine the default IP Address of the Smart-Net Gateway by analyzing the last two bytes of the MAC (Physical) address and appending them to 169.254. For example, if the MAC Address of the Smart-Net Gateway is 00:60:35:01:A0:2F then convert the hexadecimal values A0 and 2F to decimal (160 and 47) to obtain 169.254.160.47 for the IP Address.
2. Configure the PC to an address on the same network as the Smart-Net Gateway. For example: 169.254.0.1
3. Using a web browser, go to: http://169.254.xxx.xxx (where xxx.xxx are the values calculated in step 1). If the Smart-Net Gateway obtained its IP Address through DHCP, use the assigned address, instead.
4. A login page will appear. Enter **admin** for the username and **smartworks** for the password and click "Login" (Figure 3).

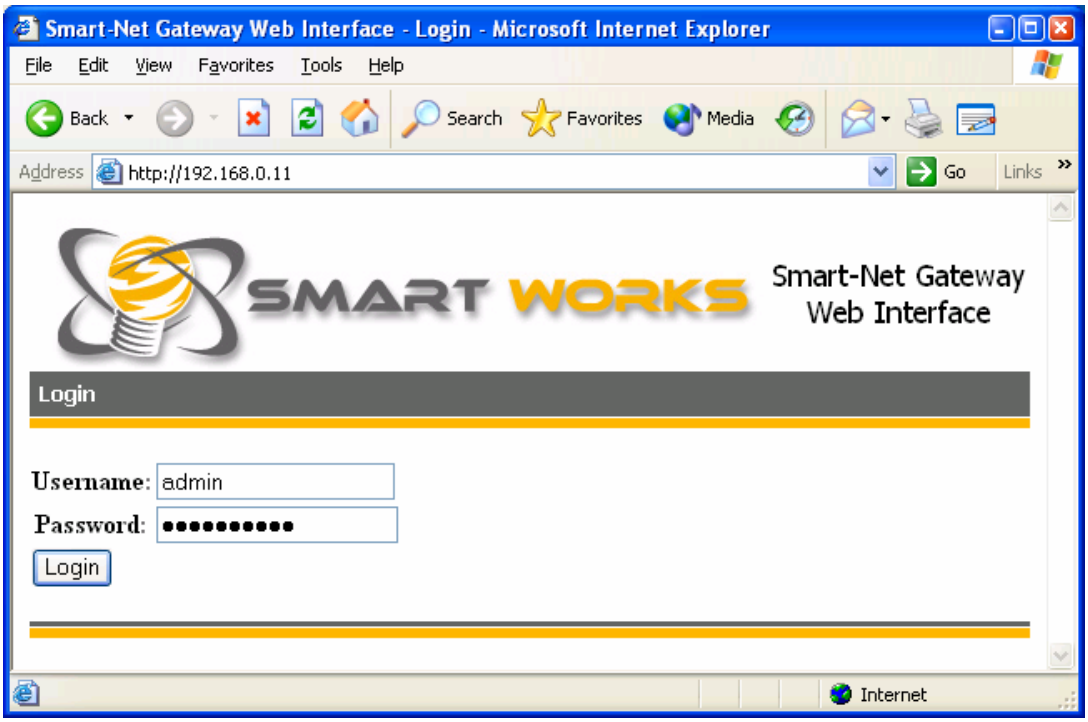

Figure 3. Web Interface Login Prompt

5. Click “System Settings” in the Menu on the left-side of the screen.
6. Enter new values for the IP Address, Subnet Mask, Default Gateway, and DNS Servers (Figure 4). Enable or Disable DHCP as needed for your network.

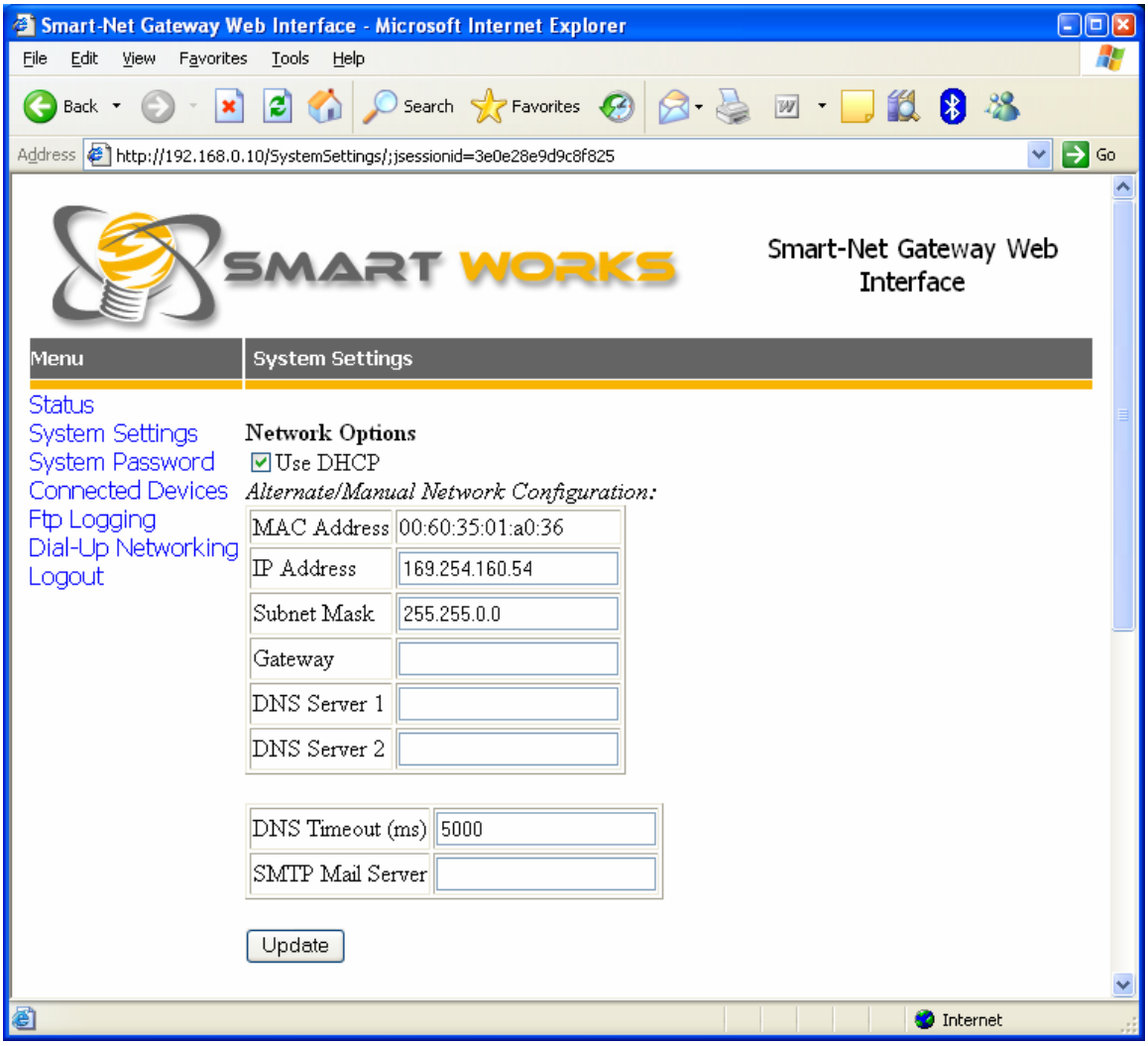

Figure 4. Web Interface System Configuration Page

7. Click “Update”.
8. When you see the text, “Saving Settings… Please Wait”, your browser may time out if you changed the IP Address to an unreachable network. This is normal and you can proceed to the next step.
9. Change the PC back to an IP Address on the same network as you just configured and proceed with configuration from the “Web Interface” section of this manual.

Web Interface

The Smart-Net Gateway provides a user-friendly web-based interface to change and view its configuration. The web interface is password-protected to only allow authorized users to make changes.

Logging in

To log into the Web Interface, open a Web browser, such as Microsoft Internet Explorer, and visit: http://xxx.xxx.xxx.xxx/, where xxx.xxx.xxx.xxx is the IP Address of the Smart-Net Gateway. If you have not yet configured an IP Address for the Smart-Net Gateway, refer to the Installation section of this manual.

When the web browser prompts for a username and password, enter **admin** for the username and **smartworks** for the password (all lower-case) if you have not yet changed the password.

Once logged in, the "Status" screen will appear, giving a brief summary of the Smart-Net Gateway's configuration (Figure 5).

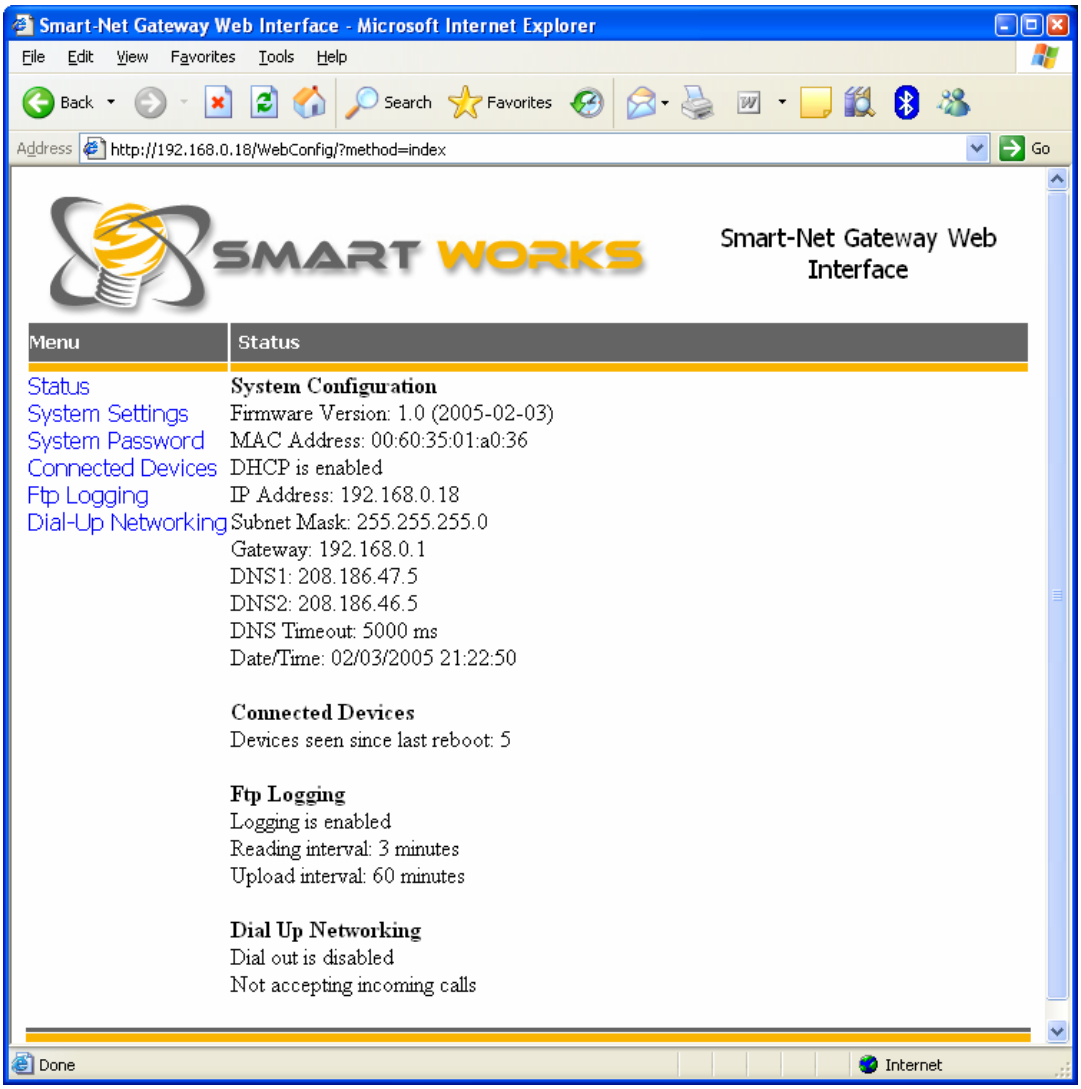

Figure 5. Web Interface Status Screen

Changing Network Settings

To change the Network Settings, click “System Settings” on the menu on the left-hand column. Make changes to the values as needed and click the “Update” button to apply the changes. Refer to Figure 4. The network settings are described below:

- **Use DHCP** – If this box is checked, the Smart-Net Gateway will attempt to retrieve an IP Address by broadcasting to a DHCP server. If the DHCP server is unreachable, the alternate/manual network configuration will be applied until an IP is successfully leased. If this box is not checked, the alternate/manual network configuration is always used.
- **MAC Address** – This value cannot be changed and represents the current Physical (MAC) Address of the network interface. This value is unique to every Smart-Net Gateway.

- **IP Address** – This value represents the TCP IPv4 Address of the Smart-Net Gateway when not using DHCP.
- **Subnet Mask** – This value represents the subnet mask associated with the IP Address of the Smart-Net Gateway when not using DHCP.
- **Gateway** – This is the IP Address of the default gateway or router to let the Smart-Net Gateway access an external network or the Internet when not using DHCP.
- **DNS Server 1** – This is the IP Address of the primary Domain Name Server to resolve domain names when not using DHCP.
- **DNS Server 2** – This is the IP Address of the backup Domain Name Server to resolve domain names if the primary Domain Name Server is inaccessible when not using DHCP.
- **DNS Timeout** – This is the number of milliseconds to wait before giving up on trying to resolve a domain name. Setting this value to zero indicates that a fallback and retry procedure should take effect. In this case, the Smart-Net Gateway will retry starting at 2 seconds and doubling until reaching 16 seconds. The Smart-Net Gateway will try four times in this period to resolve the name. * 2s * 4s * 8s * 16s. The default value for this option is 5000 ms.
- **SMTP Mail Server** – This is the IP Address or hostname of a mail server that allows the Smart-Net Gateway to send alerts and notifications via e-mail.

Adjusting the Date and Time

To change the date and time settings of the Smart-Net Gateway, click “System Settings” on the menu on the left-hand column. Below the Network Settings is a section for the “Date and Time Options” and the current date and time of the Smart-Net Gateway will be displayed (Figure 6). When you are finished setting the date and time options, click the **Update Time Options** button at the bottom to save and apply the new settings.

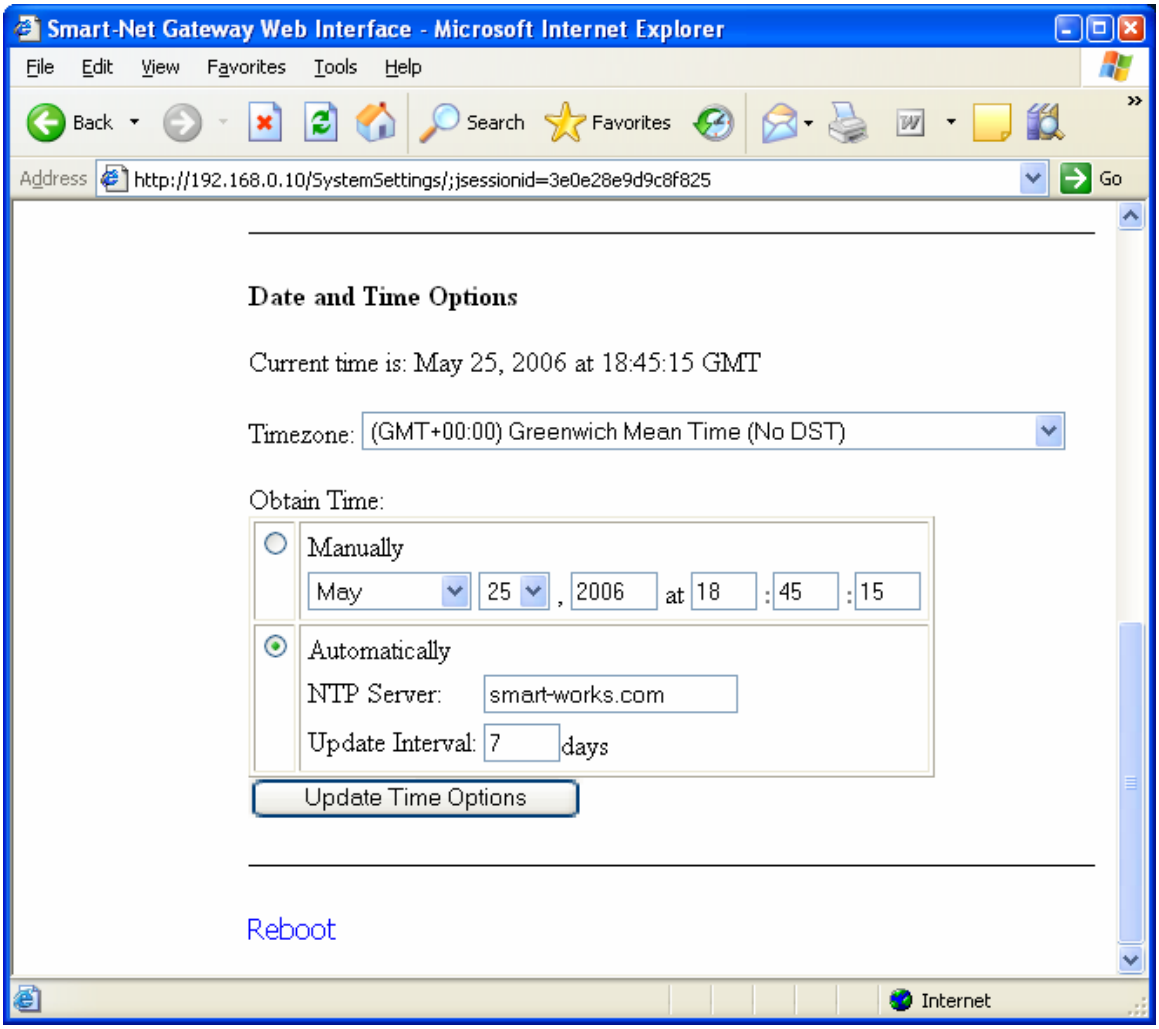

Figure 6. Web Interface Date and Time and Reboot Option

Note: When the date and time is changed or updated (either manually or automatically) on the Smart-Net Gateway by a significant amount, all connected Smart-Watt devices will have their previous reading data cleared. This ensures that an average Watt consumption calculation will be accurate since this calculation is based on timestamps that are stored on the devices.

Setting the Time Zone

The "Timezone" drop down list allows you to configure the timezone setting of the Smart-Net Gateway. By default, it is set to GMT. This time zone is used to adjust the clock for daylight saving time (DST) when necessary and save the appropriate time zone offset in the log files.

Manually Setting the Date and Time

To set the date and time of the Smart-Net Gateway manually, select "Manually", enter a date and time into the provided fields in 24-hour format, then click the **Update Time Options** button. When the page reloads, the top will inform you if the update was

successful, and the new time will be displayed in the "Date and Time Options" section. It is only recommended to use this option if the Smart-Net Gateway will not have access to a time server.

Automatically Setting the Date and Time

To set the date and time of the Smart-Net Gateway automatically, select "Automatically". In the **NTP Server** text box, enter the IP Address or hostname of a network time protocol server to use. In the **Update Interval** box, enter the frequency (in days) that the time server should be polled and the time synchronized. When finished, click the **Update Time Options** button to save the changes and set the time. When the page reloads, the top will inform you if the automatic time update was successful, and the new time will be displayed in the "Date and Time Options" section.

When this option is selected, the Smart-Net Gateway will attempt to resynchronize its clock with the specified time server during each boot and each time the **Update Time Options** button is selected, then periodically thereafter according to the update interval setting.

By default, the Smart-Net Gateway will attempt to obtain its time automatically every 7 days from "smart-works.com".

Rebooting the Smart-Net Gateway

To reboot the Smart-Net Gateway, click "System Settings" on the menu on the left-hand column. At the bottom of the page, there is a link to Reboot (Figure 6). The Smart-Net Gateway will take about one minute to reboot, during which time it is unavailable to the network. Rebooting the Smart-Net Gateway is functionally equivalent to removing and reapplying power, and does not erase any settings or log files. If DHCP is enabled, remember that the IP Address might change depending on the DHCP server's configuration.

Changing the System Password

Click the "System Password" link on the menu on the left-hand column (Figure 7). Type in the old password (**smartworks** if it has not yet been set) in the "Current Password" field and the new password in the "New Password" and "Verify New Password" fields. Then click "Submit" to apply the changes.

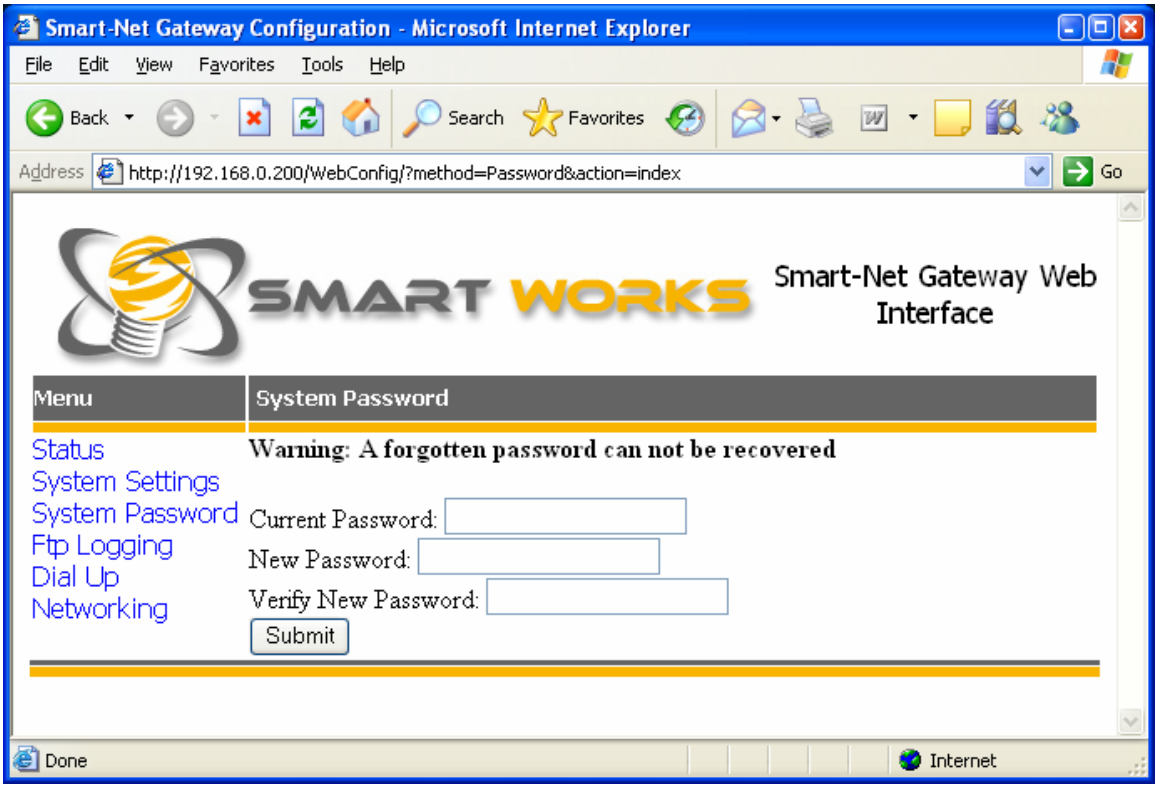

Figure 7. Changing the System Password with the Web Interface

Remember, a forgotten password can not be recovered! If you have forgotten the password, you must reset the Smart-Net Gateway to factory defaults.

Viewing the Connected Devices History

To view a table of the history of connected devices, click the “Connected Devices” link on the menu on the left-hand column. The Connected Devices History shows a table of Smart-Net devices the FTP Logger function has logged since the last time the Smart-Net Gateway was powered on or the list was cleared (Figure 8). The Connected Device History table provides a convenient way to view the Smart-Net ID, Smart-Net Device Type, UserID, and the Date/Time the device was last seen on the network. The oldest devices show up at the top of the list to help troubleshoot unresponsive devices.

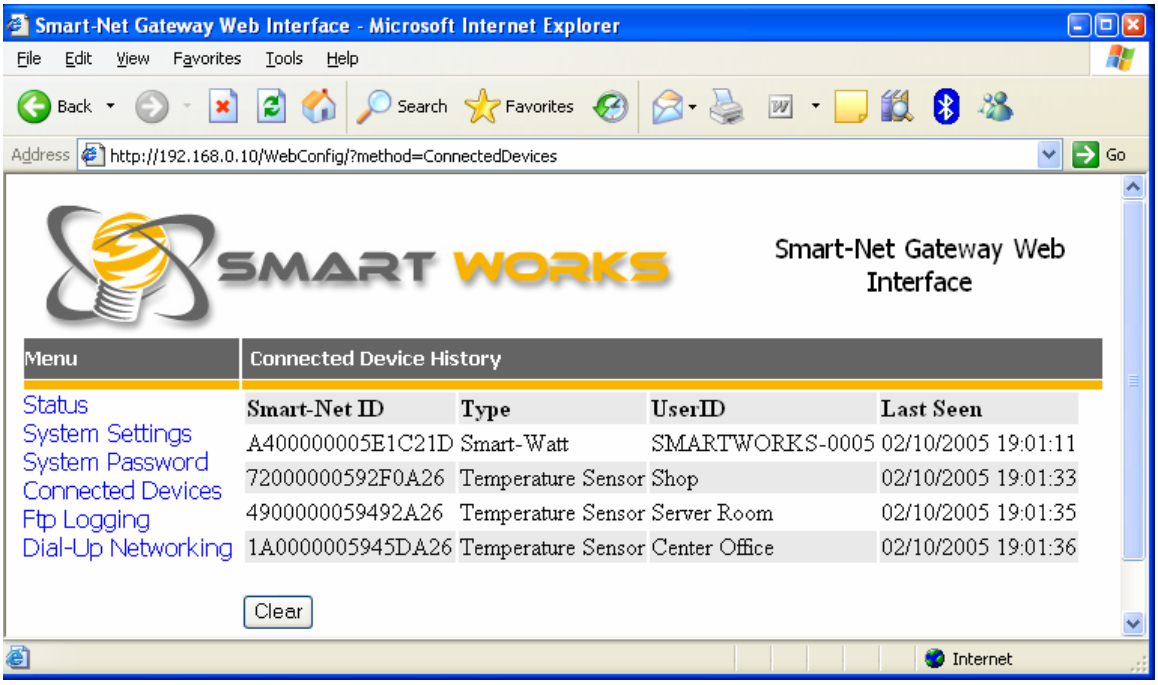

Figure 8. Connected Device History

To clear the Connected Device History, press the Clear button below the table. If the FTP Logging function has not seen any devices, a "No Devices Found" message will be displayed instead of a table. The FTP Logging function must be enabled for this table to be updated.

Changing FTP Logging Settings

To change the options for FTP Logging, click the "Ftp Logging" link on the menu on the left-hand column (Figure 9). Make the changes necessary and click "Update" or "Test" to apply the changes. For details on FTP Logging and its options, refer to the FTP Logging section of this manual.

Changing Dial Up Networking Settings

Note: Dial Up Networking support requires a Smart-Net Gateway with a modem and this option may not be present in devices without this feature.

To change options for Dial Up Networking support, click the "Dial Up Networking" link on the menu on the left-hand column (Figure 10). Make the necessary changes and click "Update" to apply the changes. For details on Dial Up Networking support, refer to "Attaching a Modem/PPP" of the FTP Logging section of this manual.

XML-RPC

Overview

The Smart-Net Gateway features an XML-RPC server that accepts commands from remote systems. Issuing XML-RPC commands to the Smart-Net Gateway provides a simple and platform independent way to remotely manage the Smart-Net Gateway and devices on the Smart-Net Network.

XML-RPC (http://www.xml-rpc.com/) was designed by UserLand Software (http://www.userland.com/) in an effort to ease cross-platform remote procedure calls. XML-RPC client implementations are available at the XML-RPC website.

XML-RPC uses the Extended Markup Language (XML) to execute Remote Procedure Calls (RPC) using the HyperText Transfer Protocol (HTTP). Because XML-RPC relies on XML for describing its procedure calls, its data can be read on any platform that is capable of parsing XML documents. Additionally, since it uses HTTP as its transport protocol, it can easily pass through most firewalls and is already implemented on many platforms. Combined, this makes it easy to implement XML-RPC support to nearly any platform that supports internet access and XML.

The descriptions for the XML-RPC Requests in this section are documented with the following notation:

<> denotes an array

> Array elements are separated by a comma. Three dots (…) indicates that more elements that follow the same pattern can be present. A vertical pipe symbol ('|') indicates that either the element to the left of or to the right of (not both) the '|' symbol may be specified.

{} denotes a struct

> Structure members are separated by a comma. Each name/value pair is separated with an '=' (equal) sign.

[] denotes an optional argument

> Square brackets indicate that the enclosed data is optional. It may be an optional parameter to a method or it may be a response that is not received in some situations.

() indicates the data type

> Possible data types are: integer, double, boolean, string, and date.

Accessing XML-RPC Methods

The Smart-Net Gateway's XML-RPC server listens to the /XmlRpc/ subdirectory of the web server on port 80. To access the XML-RPC Methods described below, send a call to the URL: **http://xxx.xxx.xxx.xxx:80/XmlRpc/** where "xxx.xxx.xxx.xxx" is the IP Address of the Smart-Net Gateway. Within several seconds, the Smart-Net Gateway will send a response, depending on the method called.

Common Field Descriptions

Several of the XML-RPC methods described below share the same fields between each other. For brevity, the descriptions of these fields are provided here instead of duplicated in the descriptions of each XML-RPC method.

Timestamp

The "Timestamp" (integer) field represents the time of the Smart-Net Gateway at the time of the method request. This field is an integer value of the number of seconds since January 1, 1970 at 00:00:00. Since the Smart-Net Gateway has no concept of time-zones, this value will normally be expressed in Universal Time Coordinated (UTC) or Greenwich Mean Time (GMT).

Status

The "Status" (integer) field is returned for all methods that require Smart-Net communications. This field is a series of bits used as flags to help troubleshoot Smart-Net or protocol problems. If this field is zero (0), this indicates that no Smart-Net problems occurred during the method call. For any other value, convert the integer to a 32-bit binary number and observe the bits that are set (logical 1). Bits 0-7 are related to Smart-Net communications. Bits 8-15 are related to the XML-RPC methods in general. Bits 24-31 depend on the specific method called. If a specific method makes use of bits 24-31, a description will be provided under that method's description.

Bit Number	**Error Description**
0	Internal Error Occurred
1	Smart-Net Port Adapter was not detected
2	Smart-Net is shorted or improperly wired
3	Communications, device, or Signal Error on Smart-Net
4	Requested Device/Parent was not found
5	Requested Device/Parent is of the wrong type or was not a Parent
6	(Reserved)
7	(Reserved)
8	Invalid parameter(s) supplied
9	Too many parameters supplied
10	Too few parameters supplied
11	Required parameter not supplied
12	(Reserved)
13	(Reserved)
14	(Reserved)
15	(Reserved)

The Bit Number referred to in the table is counted from the least significant bit (0) to the most significant bit (7). For example, if the integer returned is "4", its binary representation would be: 0000 0000 0000 0100. Since bit number 2 is set, it is determined that the Smart-Net is shorted or improperly wired.

ROMID

The “ROMID” field (string) is the Smart-Net identification string that is unique to every Smart-Net device. Knowing the ROMID is the only way to read or reconfigure a Smart-Net device. If the ROMID is not known, the **SmartNet.Find** method can be used to discover devices on the network.

ParentID

The “ParentID” field (string) is the ROMID of a parent device that contains another device. The only way to access a device that is behind a parent device is to know both the Smart-Net identification string of the device (ROMID) and the Smart-Net identification string of its parent device (ParentID).

For example, assume there is a Blade Meter with a ROMID of “2B00000005E0861D-2” connected to a socket of a SmartPDU with a ROMID of “BF0000000101741F”. To access the Blade Meter, the ROMID field would have to be “2B00000005E0861D-2” and the ParentID field would be “BF0000000101741F”.

The **SmartNet.Find** method can be used to discover devices and their parent’s on the network.

UserID

The “UserID” field (string) is a user-programmable value that can be stored in most Smart-Net devices. The maximum length of this field depends on the physical capabilities of the Smart-Net device being accessed. The UserID stored in a device can be used, for example, to help identify the physical location of a device or the name of a server that the device is attached to. The UserID is not a replacement for knowing the ROMID or a ParentID in the XML-RPC Methods.

Smart-Net Gateway XML-RPC Methods

SmartNet.Status

Parameters: <>

Response: { Timestamp=(integer), [FtpLogFileSize=(integer)], [FreeMemory=(integer)], [Uptime=(integer)], [IpAddress=(string)], [MacAddress=(string)], [FirmwareVersion=(string)], [FirmwareBuild=(string)] }

Field Descriptions:

- **FtpLogFileSize** – An integer representing the current size of the log files used by the FTP Logging feature, in bytes.
- **FreeMemory** – An integer representing the amount of free memory in the Smart-Net Gateway, in bytes.
- **Uptime** – An integer representing the number of seconds since the Smart-Net gateway has been rebooted.

- **IpAddress** – A string value of the IP Address of the SmartNet Gateway in dotted decimal notation.
- **MacAddress** – A string value of the Ethernet MAC Address of the SmartNet Gateway in the form of xx:xx:xx:xx:xx:xx where "xx" are hexadecimal values.
- **FirmwareVersion** – A String value that indicates the version of the Smart-Net Gateway's software release.
- **FirmwareBuild** – A String value that indicates the build date of the Smart-Net Gateway's software release.

The SmartNet.Status method returns general status information on the Smart-Net Gateway. The optional fields in the response structure might not be returned if the information is not applicable to the Smart-Net Gateway's current configuration. This method only takes an empty vector for parameters.

SmartNet.Time

Parameters: <[Timestamp(integer) | NTPServer(string)]>

Response: Timestamp(integer)

Field Descriptions:

- **Timestamp** – An integer representing the number of seconds elapsed since midnight on January 1, 1970.
- **NTPServer** – A string value of the IP Address or domain name for a Network Time Protocol (NTP) server to obtain the time from.

The SmartNet.Time method will set the time on the Smart-Net Gateway. The method will take an integer parameter, a string parameter, or an empty array.

When supplied with an integer, the real time clock on the Smart Net IP Gateway is updated according to the number of seconds since January 1, 1970. It is recommended that the supplied time be in Universal Time Coordinated (UTC).

When supplied with a string value representing the host name or IP Address of a Network Time Protocol (NTP) server, the Smart-Net gateway will retrieve the time from that server. If a connection failed to the NTP server, the time will not get updated and the time returned will represent current time of the Smart-Net Gateway.

In both cases, and when an empty vector is supplied, an integer value representing the number of seconds since 1970 is returned.

SmartNet.Find

Parameters: <[{[DeviceType=(string)], [ParentID=(string)]}, …]>

Response: <Timestamp (integer), Status (integer), [{ROMID=(string), [ParentID=(string)], DeviceType=(string)}, …]>

Field Descriptions:

- **DeviceType** – The type of device (i.e. "SmartWatt" or "SmartSenseTH").

Status Bits:

Bit Number	Description
24	Unrecognized DeviceType Specified

The SmartNet.Find method is used to discover what devices are connected to the Smart-Net network. Different combinations of parameters may be supplied to this method to determine how to conduct the search on the network.

If an empty vector is supplied, or a struct with no parameters, all Smart-Net Devices on the network are returned **except devices behind a parent** (i.e. Blade Meters inside a SmartPDU).

If a struct with the ParentID field is supplied, only Smart-Net Devices contained within that parent are returned.

If a struct with the DeviceType field is supplied, only devices of that type will be returned (all other types will be filtered).

If a struct with both a ParentID and DeviceType field is supplied, then only devices that match the supplied DeviceType behind the parent will be returned.

Multiple structs can be supplied to search for multiple device types and/or multiple within multiple parents in a single method call.

This method returns a vector with a timestamp and status codes followed by the results of the search as structs. Each result will contain the ROMID of the Smart-Net device found, its device type, and its ParentID (if this device was behind a parent and the request parameters indicated to search behind one or more parents).

The value of DeviceType field is the same that would be used to read or write to a device (as described in this document). For example, if the DeviceType for a device is "SmartWatt", the XML-RPC method to read that device would simply be "SmartWatt.Read".

SmartSenseTH.Read

Parameters: < ROMID(string) | { ROMID=(string), [ParentID=(string)] }, … >

Response: <Timestamp (integer), Status (integer),
{ ROMID=(string), [ParentID=(string)], [UserID=(string)], [TRH=(string)],
[Temperature=(double)], [DewPoint=(integer)], [AlertEnabled=(boolean)],
[LowTemperature=(double)], [HighTemperature=(double)],
[LowTRH=(integer)], [HighTRH=(integer)], [LowDewPoint=(integer)],
[HighDewPoint=(integer)], [InvalidKey=(boolean)] }, … >

Field Descriptions:

- **TRH** – A double value representing the true relative humidity as a percentage.
- **Temperature** – A double value representing the temperature in degrees Celsius.
- **DewPoint** – An integer value representing the dew point in degrees Celsius.

- **AlertEnabled** – A boolean value that indicates if this device is allowed to generate alerts for temperature, humidity, and dew point.
- **LowTRH** – An integer value representing the low warning true relative humidity value stored in the device as a percentage.
- **HighTRH** – An integer value representing the high warning true relative humidity value stored in the device as a percentage.
- **LowTemperature** – A double value representing the low temperature warning value stored in the device in degrees Celsius.
- **HighTemperature** – A double value representing the high temperature warning value stored in the device in degrees Celsius.
- **LowDewPoint** – An integer value representing the low dew point warning value stored in the device in degrees Celsius.
- **HighDewPoint** – An integer value representing the high dew point warning value stored in the device in degrees Celsius.
- **InvalidKey** – If the device has an invalid key, this field will be returned with a true value.

The SmartSenseTH.Read method is used to read the current environmental status from one or more Smart-Sense Temperature or Humidity sensors attached to the Smart-Net network.

Any combination of up to 10 strings and/or structs may be supplied in the parameters vector. To read a sensor, the ROMID of the sensor must be known, and, if necessary, the ParentID. To discover sensors, use the SmartNet.Find method.

The information returned is a vector whose first two elements are always integer values representing the time and a status code, respectively. The third and following elements, if present, contain a struct that contains the detailed information about each SmartSense Temperature/Humidity Sensor that was read. Any fields that are not supported by the SmartSense device will not be returned. Additionally, if the device could not be read for any reason, only the ROMID and ParentID (if supplied) will be returned for that device.

SmartSenseTH.Write

Parameters: < { ROMID=(string), [ParentID=(string)], [UserID=(string)], [AlertEnabled=(boolean)], [LowTemperature=(double)], [HighTemperature=(double)], [LowTRH=(integer)], [HighTRH=(integer)], [LowDewPoint=(integer)], [HighDewPoint=(integer)] }, … >

Response: < Status (integer), { ROMID=(string), [ParentID=(string)], Success=(boolean) }, … >

Field Descriptions:

- **Success** – A boolean that indicates if the device was updated on the network.

- **AlertEnabled** – A boolean value indicating if this device is allowed to generate alerts for temperature, humidity, and dew point.
- **LowTRH** – An integer value representing the low warning true relative humidity value stored in the device as a percentage.
- **HighTRH** – An integer value representing the high warning true relative humidity value stored in the device as a percentage.
- **LowTemperature** – A double value representing the low temperature warning value stored in the device in degrees Celsius.
- **HighTemperature** – A double value representing the high temperature warning value stored in the device in degrees Celsius.
- **LowDewPoint** – An integer value representing the low dew point warning value stored in the device in degrees Celsius.
- **HighDewPoint** – An integer value representing the high dew point warning value stored in the device in degrees Celsius.

The SmartSenseTH.Write method will write data into SmartSense Temperature or Humidity devices.

Each supplied struct in the parameters vector represents a single SmartSense Temperature or Humidity sensor device according to the ROMID and the data to store in that device.

The response is a vector whose first element is an integer value representing the time. The second and following elements, if present, contain a struct for each SmartSense device that was specified in the parameters vector. If the success boolean in the struct was true, it indicates that all the supplied data was successfully saved to the device's memory. Otherwise, if the data could not be saved for any reason, the success boolean will be false. For example, attempting to save humidity or dew point values to a temperature-only sensor will result in a success code of "false".

SmartWatt.Read

Parameters: < ROMID(string) | { ROMID=(string), [ParentID=(string)], [UpdateHistory=(boolean)] }, … >

Response: <Timestamp (integer), Status (integer), { ROMID=(string), [ParentID=(string)], [Location=(integer)], [UserID=(string)], [Offset=(integer)], [WattHr=(double)], [PwrCycle=(integer)], [ModelNumber=(string)], [InvalidKey=(boolean)], [AvgWatts=(integer)], [PrevReadTime=(integer)], [PrevWattHr=(double)], [PrevPwrCycle=(integer)], [PeakWatts=(integer)], [PeakTime=(integer)], [AlertLow=(integer)], [AlertHigh=(integer)], [AlertEnabled=(boolean)], [CircuitVoltage=(integer)], [CircuitAmperage=(integer)], [CircuitPhase=(integer)], [CircuitGroup=(integer)], [Resolution=(integer)] }, … >

Field Descriptions:

- **UpdateHistory** – A boolean that indicates whether or not this request should update the previous reading data stored for XML-RPC in the meter. This value defaults to true if not supplied.
- **Location** – If the meter is contained in a SmartPDU, this integer represents where in the SmartPDU this meter was as of the last SmartPDU.Map request.
- **Offset** – An integer representing the number of seconds elapsed since the Timestamp to determine the precise time when this particular SmartWatt device was read.
- **WattHr** – A floating-point value (double) of the cumulative Watt-Hour counter in the SmartWatt device. This value is always returned in Watt-hours and is automatically adjusted based on the Resolution field.
- **PwrCycle** – An integer value of the cumulative number of power interruptions detected by the SmartWatt device
- **ModelNumber** – A string value that contains the ModelNumber of the SmartWatt device.
- **InvalidKey** – If the device has an invalid key, this field will be returned with a true value.
- **AvgWatts** – An integer in Watts of the average watt consumption from PrevReadTime to Timestamp + Offset (now).
- **PrevReadTime** – An integer indicating the number of seconds since January 1, 1970 that this meter was last read by an XML-RPC request.
- **PrevWattHr** – An integer value indicating the cumulative Watt-Hour counter in the device at the time of PrevReadTime. This value is always returned in Watt-hours and is automatically adjusted based on the Resolution field.
- **PrevPwrCycle** – An integer value indicating the cumulative number of power interruptions detected by the SmartWatt device at the time of PrevReadTime.
- **PeakWatts** – An integer value in Watts of the highest load this meter has seen during XML-RPC reads since the peaks were last cleared.
- **PeakTime** – An integer indicating the number of seconds since January 1, 1970 that PeakWatts was last updated.
- **AlertLow** – An integer indicating the Low Alert setting, in Watts, stored in the SmartWatt device.
- **AlertHigh** – An integer indicating the High Alert setting, in Watts, stored in the SmartWatt device.
- **AlertEnabled** – A boolean value indicating if this device is allowed to generate alerts for Watt calculations and power-cycle events.
- **CircuitVoltage** – A user-programmable field that represents the rated voltage of the circuit this meter is connected to.

- **CircuitAmperage** – A user-programmable field that represents the maximum rated current capabilities that this meter is connected to.
- **CircuitPhase** – A user-programmable field that represents which phase this meter is connected to.
- **CircuitGroup** – A user-programmable field indicating which circuit group this meter is a participant in to allow for multiple meters on individual branches to be grouped automatically by reporting software to provide summary information.
- **Resolution** – An integer representing the current resolution setting of the meter. If this value is not matched to the physical properties of the meter, WattHr, AvgWatts, PrevWattHr, and PeakWatts will not be correct.

The SmartWatt.Read method is used to read power and device information from one or more SmartWatt devices connected to the Smart-Net network.

Any combination of up to 10 strings and/or structs may be specified on the parameters vector. To read a meter, the ROMID of the meter must be known, and, if necessary, the ParentID. To discover meters, use the SmartNet.Find method.

If the UpdateHistory boolean is supplied for a meter to be read and is set to "true", history information will be saved for this reading. This includes storing the current WattHr readings and the date into the meter's memory. These stored values will then be returned on the next XML-RPC call as the previous values (i.e. PrevWattHr and PrevReadTime). Additionally, the PeakWatts and PeakTime fields will be updated as necessary. If the boolean is set to false, this history data will not be saved. If unspecified, the history is automatically updated. Note that the history data only affects XML-RPC calls and has no effect on readings saved by the FTP Logging function.

The information returned is a vector whose first two elements are always integer values representing a time and a status code, respectively. The third and following elements, if present, contain a struct that contains detailed information about each SmartWatt device it read.

If a SmartWatt device could not be read for any reason, then a struct for that particular device will be returned with no information except the ROMID and ParentID (if supplied).

The Timestamp, PrevReadTime, and PeakTime fields returned are an integer value representing the number of seconds since January 1, 1970. The Offset value returned in each struct is the number of seconds elapsed since Timestamp it took to read the particular SmartWatt's Watt-Hour Counter. Therefore, Timestamp + Offset represents the exact time a specific meter was read.

PrevReadTime, PrevWattHr, PrevPwrCycle, PeakWatts, and PeakTime are the only fields specific to XML-RPC and are not affected by the FTP Logging function.

The WattHr and PrevWattHr fields are always returned as a double in Watts, regardless of the Resolution field. If the Resolution field is not set correctly, WattHr and PrevWattHr as well as all watt calculations will be inaccurate.

The Location field is automatically updated by the FTP Logging function. If FTP Logging is not enabled, use the SmartPDU.Map method to redetect and update the location field. If the SmartWatt is not behind a SmartPDU, the Location field will not be present.

CircuitVoltage, CircuitAmperage, CircuitPhase, and CircuitGroup are all user-programmable fields to help profile meters. These values are NOT automatically read nor detected by the meter.

SmartWatt.Write

Parameters: < { ROMID=(string), [ParentID=(string)], [UserID=(string)], [AlertLow=(integer)], [AlertHigh=(integer)], [AlertEnabled=(boolean)], [CircuitVoltage=(integer)], [CircuitAmperage=(integer)], [CircuitPhase=(integer)], [CircuitGroup=(integer)], [Resolution=(integer)], [ClearPrevReadings=(boolean)], [ClearPeaks=(boolean)] }, … >

Response: < Status (integer), { ROMID=(string), [ParentID=(string)], Success=(boolean) }, … >

Field Descriptions:

- **Success** -- A boolean that indicates if the device was updated on the network.
- **AlertLow** – An integer indicating the Low Alert setting, in Watts, stored in the SmartWatt device.
- **AlertHigh** – An integer indicating the High Alert setting, in Watts, stored in the SmartWatt device.
- **AlertEnabled** – A boolean value indicating if this device is allowed to generate alerts for Watt calculations and power-cycle events.
- **CircuitVoltage** – A user-programmable field that represents the rated voltage of the circuit this meter is connected to.
- **CircuitAmperage** – A user-programmable field that represents the maximum rated current capabilities that this meter is connected to.
- **CircuitPhase** – A user-programmable field that represents which phase this meter is connected to.
- **CircuitGroup** – A user-programmable field indicating which circuit group this meter is a participant in to allow for multiple meters on individual branches to be grouped automatically by reporting software to provide summary information.
- **Resolution** – An integer representing the current resolution setting of the meter. If this value is not matched to the physical properties of the meter, WattHr, AvgWatts, PrevWattHr, and PeakWatts will not be correct.
- **ClearPrevReadings** – A boolean that indicates whether or not to clear the previous reading data from the memory for the device for XML-RPC. If unspecified, this defaults to “false”.

- **ClearPeaks** – A boolean that indicates whether or not to clear the peak values from the memory for the device for XML-RPC. If unspecified, this defaults to "false".

The SmartWatt.Write method will write data into SmartWatt devices.

Each supplied struct in the parameters vector represents a single SmartWatt device according to the ROMID and ParentID (if necessary) and the data to store in that device.

The response is a vector whose first element is an integer value representing the time. The second and following elements, if present, contain a struct for each SmartWatt device that was specified in the parameters vector. If the success boolean in the struct is true, it indicates that all the supplied data was successfully saved to the device's memory. Otherwise, if the data could not be saved for any reason, the success boolean will be false.

If ClearPrevReadings is set to true, then previous reading counters in the device memory that is used for average watt consumption values will be cleared. In this case, the next SmartWatt.Read will not display these values.

If ClearPeaks is set to true, then the peak value in the SmartWatt will be cleared. In this case, the next SmartWatt.Read will always set a new peak value.

When setting the Resolution field, it is important to make sure to configure it to match the physical properties of the meter. If the resolution is incorrectly set, all Watt calculations and Watt-hour readings will be inaccurate.

SmartPDU.Read

Parameters: < ROMID(string), … >

Response: < Timestamp (integer), Status (integer),
{ ROMID=(string), [UserID=(string)], [Size=(integer)] }, … >

Field Descriptions:

- **Size** – An integer that represents the maximum number of individual circuits this SmartPDU supports.

The SmartPDU.Read method is used to read information of one or more SmartPDUs attached to the Smart-Net network. Note that this method does NOT read or search for any sensors or meters contained within the SmartPDU.

Up to 10 strings (ROMIDs) may be supplied in the parameters vector. To read a SmartPDU, the ROMID of the SmartPDU must be known. To discover SmartPDUs, use the SmartNet.Find method.

The information returned is a vector whose first two elements are always integer values representing the time and a status code, respectively. The third and following elements, if present, contain a struct that contains the detailed information about each that was read. If the device could not be read for any reason, only the ROMID will be returned for that device.

The Size field returned is preprogrammed by the factory into the SmartPDU and cannot be changed. The UserID field can be changed with either SmartPDU.Write or SmartSenseTH.Write since the same UserID is shared between both the SmartPDU's internal temperature sensor and the SmartPDU itself.

SmartPDU.Map

Parameters: < ROMID(string) >

Response: < Timestamp (integer), Status (integer), { ROMID=(string), Location=(integer) }, … >

Field Descriptions:

- **Location** – An integer representing the position of the meter within the SmartPDU.

The SmartPDU.Map method is used to discover the position and which terminal in the SmartPDU a particular blade meter (SmartWatt) is connected to. This method also updates the memory in the SmartWatt with the new location information, and, if the SmartWatt had no UserID set, writes a default UserID based on this location.

Only one SmartPDU may be mapped at a time. To map devices in a SmartPDU, the ROMID of the SmartPDU must be known. The ROMID of devices within a SmartPDU do not need to be known. To discover SmartPDUs, use the SmartNet.Find method.

The information returned is a vector whose first two elements are always integer values representing the time and status code, respectively. The third and following elements, if present, contain a struct that contains the ROMID and location of each device in the SmartPDU whose position was successfully detected.

It is not necessary to call this method if the FTP Logging function is enabled since the meter's location information will automatically be updated during routine reads. However, if FTP Logging is disabled and meters have been changed in a SmartPDU, or this is a new SmartPDU installation, this method must be called in order to get accurate results when reading the meters.

SmartPDU.Write

Parameters: < { ROMID=(string), [UserID=(string)] }, … >

Response: < Status (integer), { ROMID=(string), Success=(boolean) }, … >

Field Descriptions:

- **Success** – A boolean that indicates if the device was updated on the network.

The SmartPDU.Write method will write data into SmartPDU devices.

Each supplied struct in the parameters vector represents a single SmartPDU device according to the ROMID and the data to store in that device.

The response is a vector whose first element is an integer value representing the time. The second and following elements, if present, contain a struct for each SmartPDU device that was specified in the parameters vector. If the success boolean in the struct

was true, it indicates that all the supplied data was successfully saved to the device's memory. Otherwise, if the data could not be saved for any reason, the success boolean will be false.

Note that setting the UserID of a SmartPDU will also set the UserID of the SmartPDU's built in temperature sensor.

Additional Resources

For additional information regarding the implementation, benefits, and features of the XML-RPC protocol, consult the following resources:

- Programming Web Services with XML-RPC by Simon St.Laurent, Joe Johnston, and Edd Dumbill. O'Reilly, 2001. ISBN: 0-596-00119-3.
- The XML-RPC Home Page at http://www.xml-rpc.com/.
- The XML-RPC Specification at http://www.xmlrpc.com/spec.

FTP Logging

Overview

The Smart-Net Gateway has the ability to automatically read information on every Smart-Net device connected to it, store it to a log file, and upload this log file to an FTP server at a specified interval. One advantage of using FTP Logging is there is no requirement to continually communicate with the Smart-Net Gateway to obtain the latest Smart-Net Device readings and status information. Because all data logging is performed unattended and automatically, FTP Logging can be used in situations where a constant internet connection or direct internet access is not desired or feasible.

The Smart-Net Gateway will append its internal log data at the specified interval to the FTP Server. Once the data has been successfully uploaded, the internal log data is cleared to free memory for new log data.

Configuring and Testing

To configure the options for FTP Logging, log into the Web Interface as described earlier in this manual and click “Ftp Logging” in the menu column (Figure 9). Make the necessary changes, using the following section as a guide to explain each option, and click “Update” to save the changes.

To ensure that the changes work, a “Test” option is provided. This will create a text file named “xxxxxxxxxxxxxxxx-Test.txt” (where “xxxxxxxxxxxxxxxx” is the serial number of the Smart-Net Gateway) containing a log line showing the current date and time and the string “Smart-Net Gateway Test file”. This file will then be uploaded to the FTP Server. Note that the test button will upload a test file regardless of whether the FTP Logging function is enabled or disabled. If the test file fails to upload, you must either Enable Ftp Logging or click the Test button again to retry the upload.

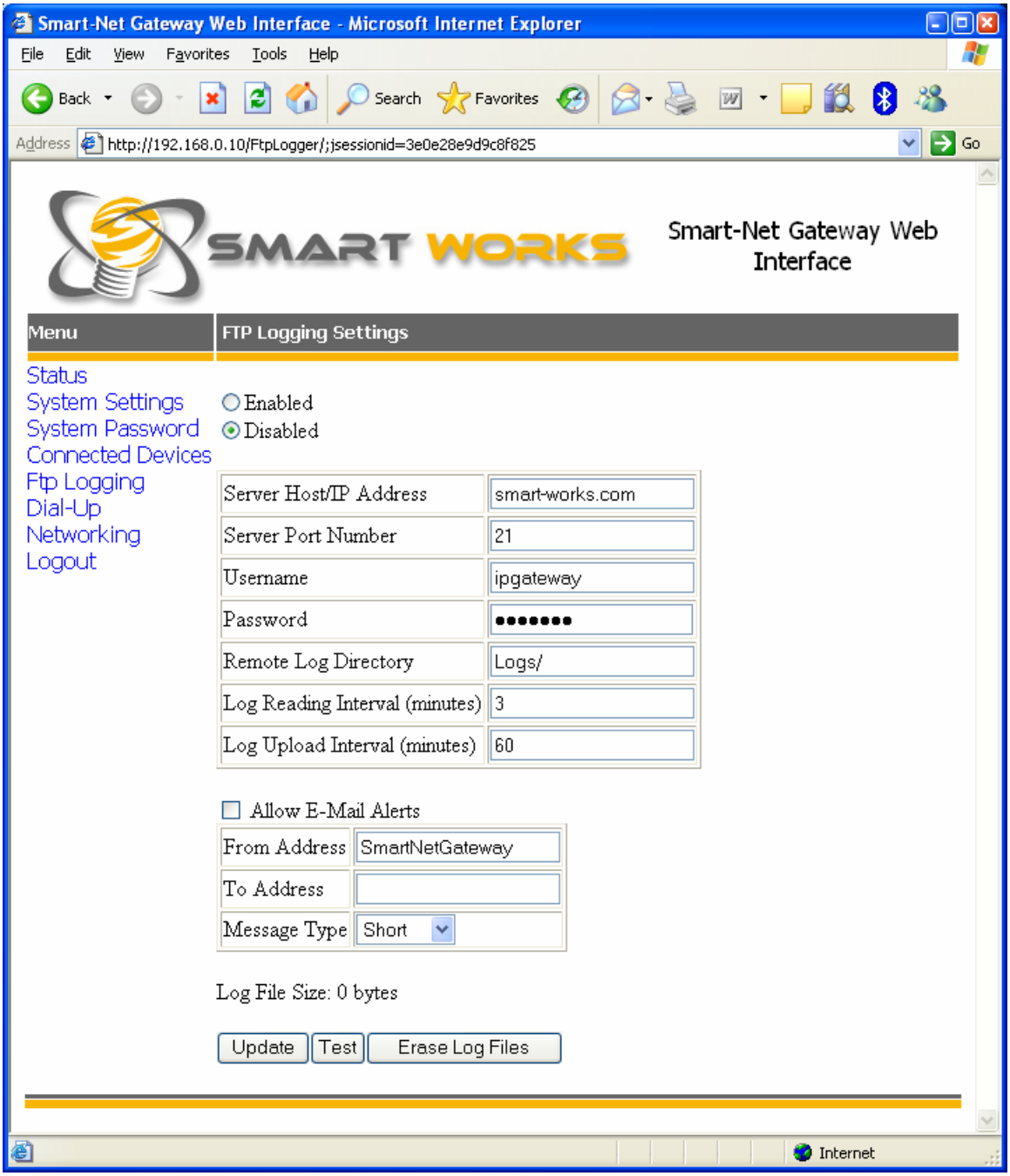

Figure 9. FTP Logging Configuration

The following describes each configurable option:

Enabling/Disabling

By default, FTP Logging is disabled. When it is disabled, the device will **not** automatically collect information on Smart-Net devices and it will **not** automatically upload to an FTP server. To enable this feature, fill in the desired values for server host, port, username, password, remote log directory, reading interval, and upload interval and click “Update” or “Test”.

Server Host

The “Server Host/IP Address” field specifies the hostname or IP address of the FTP Server in which the Smart-Net Gateway should upload its log data to. By default, it is set to “smart-works.com” for a public server.

The “Server Port Number” field specifies which port the FTP Server is listening on. By default, FTP uses port 21. This value should only be changed if your FTP Server uses a non-standard port number.

Server Username/Password

If the FTP server that the Smart-Net Gateway connects to uses an anonymous login, then “anonymous” should be entered for the username and an e-mail address for the password. If the FTP server has restricted address, then the proper username/password should be filled in here. Note that FTP is an insecure protocol and the password will be transmitted in clear-text across the network on each upload attempt.

Remote Log Directory

A directory on the FTP server to save files to may be specified here. By default, this is “Logs/” and may be case sensitive, depending on the operating system of the FTP Server.

Reading and Upload Intervals

The reading interval specifies how long to wait from the time the previous reading finished before reading all of the connected Smart-Net devices’ data again. The upload interval specifies how often to connect to the FTP Server and append this read data to the log file on the server. The default values are 3 minutes for the reading interval and 60 minutes for the upload interval.

Alerting Options

The “FTP Logging” section of the web interface also allows for configuring the settings to generate e-mail based alerts from the FTP Logging feature.

The Alerting feature will automatically send an E-Mail alert every time a Smart-Net device is considered to be in an “Alerting” or “Alarming” state. Each Smart-Net device has its own individual alert settings that are stored directly in the device. To configure an individual Smart-Net device’s settings, you may use the XML-RPC feature.

Note: In order for the Alerting feature to work, an SMTP Mail Server that allows relaying to the supplied e-mail address must be configured as described in the section “Changing Network Settings” of this manual.

Enabling/Disabling

To enable the ability to send alerts, select the checkbox marked “Allow E-Mail Alerts” and enable the FTP Logging feature as described above. To disable alerts, simply uncheck the “Allow E-Mail Alerts” box. Note that if none of the Smart-Net devices are configured to allow alerts, no alerts will be sent even if this feature is activated. By default, this feature disabled.

From Address

The “From Address” field specifies the E-Mail address that the alert messages should appear to have originated from. By default, this is “SmartNetGateway”. However, it may be desired to change it to an address to help identify the Smart-Net Gateway that the alert was sent from. For example, the “From” field may be set to:

SmartNetGateway0001

SmartNetGateway0001@example.net

Note that some mail servers might reject messages with an invalid “From” address causing alert messages to not be delivered.

To Address

The “To Address” field specifies the E-Mail address to send the alert messages to. For example, alerts@example.net may be entered. Note that some mail servers disallow relaying to foreign E-Mail addresses causing alert messages not to be delivered.

Message Type

The “Message Type” setting changes the type of E-Mail that is sent to the recipient.

Selecting “Short” will send alert messages that are very brief and suitable for delivery to a cell phone or a pager.

Selecting “Normal” will send alert messages that contain every detail about the device and the alert condition, suitable for reading in a standard e-mail client.

Log File Formats

All log file names generated by the FTP Logging feature start with the unique serial number of the Smart-Net Gateway for easy identification and end with the .txt file extension as follows:

<serial number>-<type>.txt

Where <type> is an indication of what type of log data the file represents (described below) and <serial number> is the unique serial number of the Smart-Net Gateway’s internal Smart-Net adapter that generated the log file.

The reading log files are stored in a structure designed for easy importing into a database. Most Smart-Net devices create a log file of two types: The static configuration data of each device and the changing reading data of each device. The static configuration log files are replaced on every upload and are usually named as “<devicetype>s.txt” where <devicetype> is the type of Smart-Net device. Similarly, the reading data log file is usually named as “<devicetype>Reads.txt”. For example, a SmartWatt will have log files whose filenames end with “SmartWatts.txt” and “SmartWattReads.txt”.

Common Fields

Several log files will contain the same types of fields as other log files. For brevity, fields that appear in several log files are described here instead of duplicating the same information in the subsequent sections.

`SmartNetID`	A globally unique identifier for this particular Smart-Net device (also known as the ROMID).
`SmartNetGroupID`	An identifier of which Smart-Net branch this Smart-Net device is connected to. This is the same unique address/serial number of the Smart-Net Gateway or the USB/Serial port of the Virtual Smart-Net Gateway.
`ParentID`	The SmartNetID of this device's parent if the device is contained within or is an accessory of another device (such as a Smart-PDU).
`Location`	If this device is part of a parent, this field may indicate where in the parent that this device is located and/or reading. For example, a Smart-Watt's location in a Smart-PDU will be mapped to the Terminal/Slot number that the meter is attached to.
`UserID`	A user-programmable field to provide a "friendly" name for this device. If the UserID stored in the device contains quotation marks or commas, they will be replaced by underscores before being saved to the log file.
`DateTime`	Indicates the Date and Time that this reading occurred. It is in the format of `MM/DD/YYYY HH:MM:SS`.
`TimeZone`	Indicates the GMT offset of the `DateTime` field. It is in the format of `+HH:MM` where `HH` is the number of hours and `MM` is the number of minutes forward or backward (as indicated by a `+` or `–` sign) from GMT time.

Note that all fields given in the descriptions will be present in the log file. However, if a field is not applicable to a specific device, that field will be empty (null).

Each field in a log entry is separated by only a comma (`,`) character. If the field's data type is to be treated as a string of text, it is enclosed in quotes in the log file. Each log entry is terminated by a carriage-return and a newline.

SmartSenseTH

Log files generated by Smart-Sense Temperature or Humidity sensors are named “xxxx-SmartSenseTHs.txt” and “xxxx-SmartSenseTHReads.txt”, where “xxxx” is the serial number of the Smart-Net Gateway.

SmartSenseTHs.txt

The static configuration log file, SmartSenseTHs.txt, contains the following information for each sensor:

```
SmartNetID, SmartNetGroupID, ParentID, Location, UserID,
AlertTempL, AlertTempH, AlertHumidL, AlertHumidH, AlertDewPointL,
AlertDewPointH, AlertEnabled
```

The field descriptions for SmartSenseTHs.txt are:

`Location`	This field is empty; Reserved for future use
`AlertTempL / AlertTempH`	The low or high temperature, in degrees Celsius, the sensor must meet or exceed before the Smart-Net Gateway will send an alert.
`AlertHumidL / AlertHumidH`	The low or high true relative humidity, in percent (0-100), the sensor must meet or exceed before the Smart-Net Gateway will send an alert. If this is not a humidity sensor, these fields will be empty.
`AlertDewPointL / AlertDewPointH`	The low or high dew point, in degrees Celsius, the sensor must meet or exceed before the Smart-Net Gateway will send an alert. If this is not a humidity sensor, these fields will be empty.
`AlertEnabled`	A true or false value that indicates if this device is allowed to generate alerts. When false, all temperature, humidity, and dew point alerts are disabled.

For example, one line of a sensor’s configuration may be:

```
"B600000059488826","425E700344890389",,,"RACK 2001-
A",15.0,25.0,25,75,0,7,true
```

SmartSenseTHReads.txt

The reading log file, SmartSenseTHReads.txt, contains the following reading data for each sensor:

```
SmartNetID, DateTime, TimeZone, Temp, Humid, DewPoint
```

The field descriptions for SmartSenseTHReads.txt are:

`Temp`	The temperature, in degrees Celsius, at the sensor.

`Humid`	The true relative humidity value, a percentage from 0 to 100, at the sensor. If this is not a humidity sensor, this value will be empty.
`DewPoint`	The Dew Point, as calculated from the temperature and humidity, in degrees Celsius, at the sensor. If this is not a humidity sensor, this value will be empty.

For example, one line of a sensor's log entry may be:

```
"B600000059488826",02/14/2006 20:41:44,-08:00,11.0,65.9,4.9
```

SmartWatt

Log files generated by Smart-Watts and Blade Meters are named "xxxx-SmartWatts.txt" and "xxxx-SmartWattReads.txt", where "xxxx" is the serial number of the Smart-Net Gateway.

SmartWatts.txt

The static configuration log file, SmartWatts.txt, contains the following information for each meter:

```
SmartNetID, SmartNetGroupID, ParentID, Location, UserID, Model,
AlertLow, AlertHigh, Resolution, CircuitVolt, CircuitAmp,
CircuitPhase, CircuitGroup, AlertEnabled
```

The field descriptions for SmartWatts.txt are:

`Location`	The Terminal number/CT number in the Smart-PDU this meter is attached to. If this meter is not part of a Smart-PDU, this field is empty.
`Model`	The model-number of this Smart-Watt enclosed in quotes.
`AlertLow`	The lowest number of Watts this meter can read before the Smart-Net Gateway sends an alert.
`AlertHigh`	The highest number of Watts this meter can read before the Smart-Net Gateway sends an alert.
`Resolution`	A look-up table indicating the resolution of this Smart-Watt. Note that all readings are converted to Wh in the log files regardless of the resolution.
`CircuitVolt`	A user-programmable value to indicate the Voltage of the attached circuit.
`CircuitAmp`	A user-programmable value to indicate the maximum current (in Amps) that the attached circuit will support.

`CircuitPhase`	A user-programmable value to indicate which phase this meter is attached to (typically 1, 2, or 3). This is used by reporting software to gather total readings grouped by phase.
`CircuitGroup`	If this meter is part of a group of meters to measure a multi-phase circuit, this value indicates which group this meter belongs to. This is used by reporting software to aggregate multiple meters into a single "virtual" circuit (i.e. a 3-phase circuit with three meters, one per phase). The groups are unique to each SmartNetGroupID.
`AlertEnabled`	A true or false value that indicates if this device is allowed to generate alerts. When false, the low and high Watt alerts, as well as power cycle alerts, are disabled.

For example, one line of a Smart-Watt's configuration may be:

```
"7400000005D9E61D-
1","425E700344890389","BF0000000101741F",13,"RACK 1357","SD25099-
000-0000",0,800,1,120,20,2,0,false
```

SmartWattReads.txt

The reading log file, SmartWattReads.txt, contains the following data for each meter:

```
SmartNetID, DateTime, TimeZone, WattHours, PowerCycles, Watts
```

The field descriptions of SmartWattReads.txt are:

`WattHours`	The cumulative Watt-Hour reading of this meter. Note that this value is always converted to Wh, regardless of the resolution setting.
`PowerCycles`	The cumulative number of power interruptions this meter has detected. If the meter doesn't support power cycle detection, this field will be empty.
`Watts`	The calculated load, in Watts, seen by this meter as determined from the previous reading and the current reading. If there was no previous reading to calculate from or the Smart-Net Gateway's date and time was adjusted since the previous read, this field will be empty.

For example, one line of a Smart-Watt's reading data may be:

```
"7400000005D9E61D-1",02/14/2006 22:34:22,-08:00,94183.1,,204
```

SmartPDU

The log file generated by a Smart-PDU is named "xxxx-SmartPDUs.txt", where "xxxx" is the serial number of the Smart-Net Gateway. Since the Smart-PDU itself has no changing data, there is no associated SmartPDU "Reads" file. The data of devices

contained within a SmartPDU are in the appropriate SmartWatt and SmartSenseTH log files.

The configuration log file, SmartPDUs.txt, contains the following data for each SmartPDU:

```
SmartNetID, SmartNetGroupID, UserID, Size, SmartSenseID
```

The field descriptions of SmartPDUs.txt are:

`Size`	The maximum number of circuits this PDU can support if fully populated (i.e. 42).
`SmartSenseID`	The SmartNetID of the Smart-Sense device built-in to the Smart-PDU. The readings of this sensor are logged in the appropriate SmartSenseTH log file.

For example, one line of a Smart-PDU's configuration data may be:

```
"BF0000000101741F","425E700344890389","PANEL0507",42,"7F00000066D
97626"
```

Smart-Net

The log file generated by the Smart-Net Gateway itself is named "xxxx-SmartNet.txt", where xxxx is the serial number of the Smart-Net Gateway. This file is replaced on every FTP upload and contains the configuration information of the Smart-Net Gateway as name/value pairs divided into groups.

Exception

The exception file is named "xxxx-Exception.txt" where xxxx is the serial number of the Smart-Net Gateway.

Many problems or notices that the Smart-Net Gateway's Log Collection feature encounters will be saved to the Exception log file.

Exceptions may be generated if there is a problem with the Smart-Net network at the time of a device read, an abnormal problem with the Smart-Net Gateway, or a defective Smart-Net device. Intermittent exceptions may also be generated if there is a wiring problem with the Smart-Net network, including, but not limited to, noise, low-quality cables, long distances, lose connections, adding/removing devices, etc.

The entries in the Exception.txt file have the following format:

```
DateTime, TimeZone, ExceptionType, ExceptionDescription
```

The field descriptions of the Exception.txt file are:

`ExceptionType`	The kind of exception that has occurred, enclosed in quotes. For example, if there was a problem reading a Smart-Watt, this field will be "Smart-Watt".

`ExceptionDescription`	A detailed description of the problem that occurred, enclosed in quotes.

Test

The test file is named “xxxx-Test.txt” where xxxx is the serial number of the Smart-Net Gateway. This file is uploaded when the “Test” option is selected in the configuration.

The entries in this file have the following format:

```
DateTime, TimeZone, "Smart-Net Gateway Test file"
```

The test file is used to verify that communication with the specified FTP Server can be established and is working.

Intervals and Memory Usage

The minimum recommended Log Reading Interval is three minutes. This is to ensure an accurate measurement of cumulative watt-hour readings as well as an accurate calculation of average watt consumption between readings.

The Log Upload Interval should not be set to too long of a value. This is to ensure the Smart-Net Gateway doesn’t run out of memory in large Smart-Net networks between uploads from storing the log files. If the log file memory becomes full, the Smart-Net Gateway will attempt to upload the log file immediately, regardless of the Log Upload Interval setting. If the upload fails, the data collected on that reading interval will be lost. The Smart-Net Gateway can hold over 3000 log entries. Additionally, a long upload interval will delay how quickly the log data can be imported into a remote database and processed. The default upload interval is one hour (60 minutes).

Attaching a Modem/PPP

Note: This option requires a Smart-Net Gateway that supports this feature. To determine if this feature is supported, look for the “Dial Up Networking” option in the menu of the Web Interface (Figure 10).

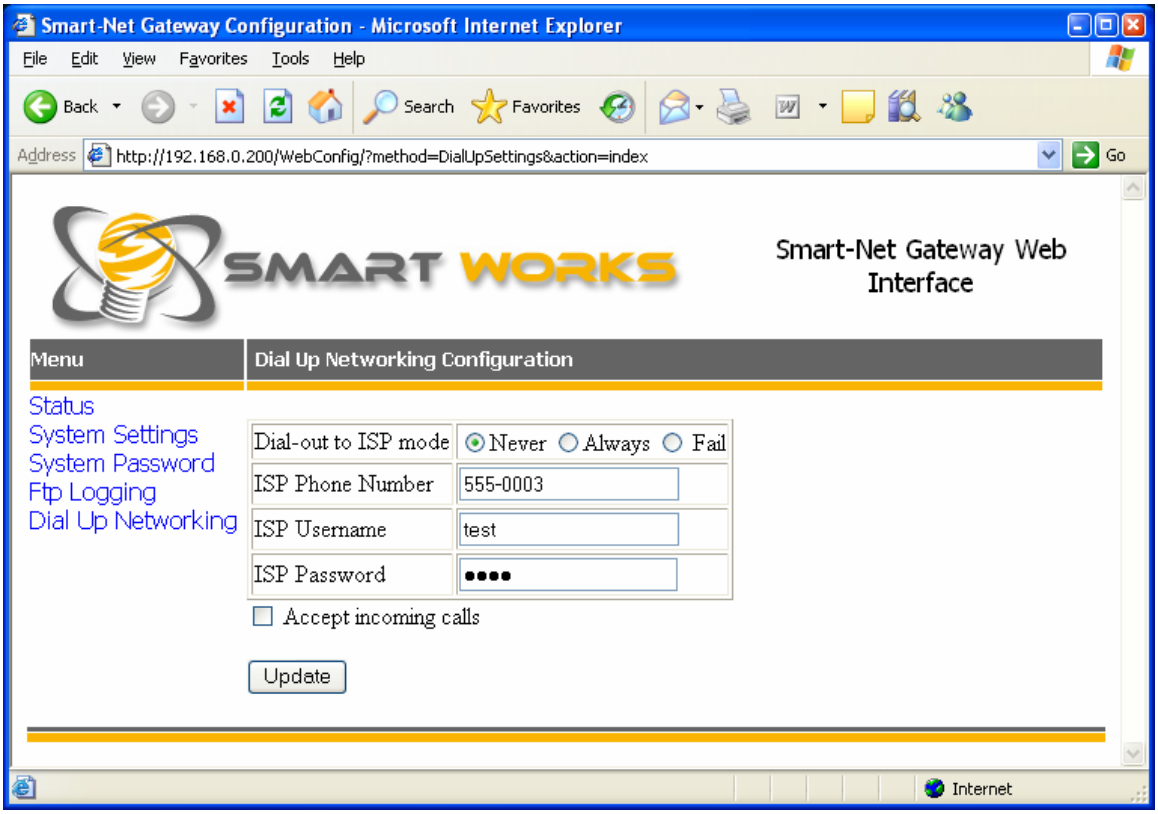

Figure 10. Dial Up Networking Configuration

An external serial modem may be connected to the DB9 Male connector on the Smart-Net Gateway to provide an alternate method of network communication for the Smart-Net Gateway.

To enable Dial Up Networking support, log in to the Web Interface and select “Dial Up Networking” from the menu. The following describes each configurable option:

- **Dial-Out to ISP Mode** – If this is set to “Never”, then the Smart-Net Gateway will not attempt to make out-bound calls for the purposes of FTP Logging. If this is set to “Always”, the Smart-Net Gateway will attempt to dial before sending any log file. If this is set to “Fail”, the Smart-Net Gateway will only attempt to dial if the Ethernet connection fails to upload the log file.
- **ISP Phone Number** – This is the phone number of the Internet Service Provider (or other PPP compatible network connection). The phone number will be dialed as-is. Extra codes, such as waiting for a second dial tone or dialing delays, are modem-specific. Consult your modem’s documentation for usage information.
- **ISP Username and Password** – The username and password used to authenticate the PPP session on the remote peer.
- **Accept Incoming Calls** – If this is checked, the Smart-Net Gateway will also answer all incoming calls. The Smart-Net Gateway will start a PPP session and will require a username and password combination (which is the same as the system username and password). The Smart-Net Gateway will assign itself an IP

Address of 192.168.0.1 and the remote peer (incoming caller) an IP Address of 192.168.0.2. These IP Addresses may be used for remote configuration or XML-RPC access.